Intrinsic Hardware Evolution for the Design and Reconfiguration of Analog Speed Controllers for a DC Motor

David A. Gwaltney
NASA Marshall Space Flight Center
Huntsville, AL 35812
David.A.Gwaltney@nasa.gov

Michael I. Ferguson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Michael.I.Ferguson@jpl.nasa.gov

Abstract

Evolvable hardware provides the capability to evolve analog circuits to produce amplifier and filter functions. Conventional analog controller designs employ these same functions. Analog controllers for the control of the shaft speed of a DC motor are evolved on an evolvable hardware platform utilizing a second generation Field Programmable Transistor Array (FPTA2). The performance of an evolved controller is compared to that of a conventional proportional-integral (PI) controller. It is shown that hardware evolution is able to create a compact design that provides good performance, while using considerably less functional electronic components than the conventional design. Additionally, the use of hardware evolution to provide fault tolerance by reconfiguring the design is explored. Experimental results are presented showing that significant recovery of capability can be made in the face of damaging induced faults.

1 Introduction

Research on the application of hardware evolution to the design of analog circuits has been conducted extensively by many researchers. Many of these efforts utilize a SPICE simulation of the circuitry, which is acted on by the evolutionary algorithm chosen to evolve the desired functionality. An example of this is the work done by Lohn and Columbano at NASA Ames Research Center to develop a circuit representation technique that can be used to evolve analog circuitry in software simulation[1]. This was used to conduct experiments in evolving filter circuits and amplifiers. A smaller, but rapidly increasing number of researchers have pursued the use of physical circuitry to study evolution of analog circuit designs. The availability of reconfigurable analog devices via commercial or research-oriented sources is enabling this approach to be more widely studied. Custom Field Programmable Transistor Array (FPTA) chips have been used for the evolution of logic and analog circuits. Efforts at the Jet Propulsion Laboratory (JPL) using their second generation FPTA2 chip are documented in [2,3,4]. Another FPTA development effort at Heidelberg University is described in [5]. Some researchers have conducted experiments using commercially available analog programmable devices to evolve amplifier designs, among other functions[6,7].

At the same time, efforts to use evolutionary algorithms to design controllers have also been widely reported. Most of the work is on the evolution of controller designs suitable only for implementation in software. Koza, et al., presented automatic synthesis of control laws and tuning for a plant with time delay using genetic programming. This was done in simulation [8]. Zebulum, et. al., have evolved analog controllers for a variety of industrially representative dynamic system models[10]. In this work, the evolution was also conducted in a simulated environment.

The ability to provide fault-tolerance via hardware evolution is a parallel interest in the efforts of many researchers. In [10], Zebulum, et. al., incorporated fault tolerance as an objective in one of their experiments. In this case, components were removed from a design, and the design was evaluated with each component missing as part of the fitness assessment. This resulted in a design that could tolerate nine different individual faults while providing slight degradation in response when compared to the non-faulty case. Canham and Tyrrell [11] incorporated fault tolerance in the evolution of a digital oscillator on a Field Programmable Gate Array (FPGA) by injection of faults during the evaluation process. This resulted in oscillator designs that can tolerate multiple simultaneous faults. Lohn, et. al. [12] have demonstrated the capability for evolution to repair state machine designs implemented in an FPGA. A random stuck-at-zero fault is introduced into the state machine design on the FPGA to create a fault, and evolution is employed to modify the design to operate correctly in the presence of the fault. Keymeulen, et. al., [13] compared two methods of achieving fault tolerance in an evolved design, one based on the fitness definition and the other based on the evolved population. This was applied to faults encountered during the operation of an XNOR and an analog multiplier on an FPTA device. In these examples, it was found that the populational approach was superior. This approach makes use of the resulting population from the evolution of the original, non-faulty circuitry, as a source of existing designs to accommodate the fault, or as a basis for further evolution.

Hardware evolution can enable the deployment of a self-configurable controller in hardware. Such a controller will be able to adapt to environmental conditions that would otherwise degrade performance, such as temperature varying to extremes or ionizing radiation. Hardware evolution can provide fault-tolerance capability by re-routing internal connections around damaged components or by reuse of degraded components in novel designs. These features, along with the capability to accommodate unanticipated or changing mission requirements, make an evolvable controller attractive for use in a remotely located platform, such as a spacecraft. Hence, this effort focuses on the application of hardware evolution to the intrinsic, or *in situ,* configuration of a shaft speed controller for a DC motor. To this end, the Stand-Alone Board-Level Evolvable (SABLE) System[3], developed by researchers at the Jet Propulsion Laboratory, is used as the platform to evolve analog speed controllers for a DC motor.

Motor driven actuators are ubiquitous in the commercial, industrial, military and aerospace environments. A recent trend in aviation and aerospace is the use of power-by-wire technologies. This refers to the use of motor driven actuators, rather than hydraulic actuators for aero-control surfaces[14][15]. Motor driven actuators have been considered for upgrading the thrust vector control of the Space Shuttle main engines [16]. In spacecraft applications, servo-motors can be used for positioning sun-sensors, Attitude and Orbit Control Subsystems (AOCSs), antennas, as well as valves, linear actuators and in other closed-loop control functions.

In this age of digital processor-based control, analog controllers are still frequently used at the actuator level in a variety of systems. In the harsh environment of space, electronic components must be rated to survive temperature extremes and exposure to radiation. Very few microcontrollers and digital signal processors are available that are rated for operation in a radiation environment. However, operational amplifiers and discrete components are readily available and are frequently applied.

Reconfigurable analog devices provide a small form factor platform on which multiple analog controllers can be implemented. The second generation Field Programmable Transistor Array (FPTA2), as part of the SABLE System, is a perfect platform for implementation of multiple controllers, because its sixty-four cells can theoretically provide sixty-four operational amplifiers, or evolved variations of amplifier topologies. Further, its relatively small size and low power requirements provide savings in space and power consumption over the uses of individual operational amplifiers and discrete components[2].

The round-trip communication time between the Earth and a spacecraft at Mars ranges from 10 to 40 minutes. For spacecraft exploring the outer planets the time increases significantly. A spacecraft with self-configuring controllers could work out interim solutions to control system failures in the time it takes for the spacecraft to alert its handlers on the Earth of a problem. The evolvable nature of the hardware allows a new controller to be created from compromised electronics, or the use of remaining undamaged resources to achieve required system performance. Because the capabilities of a self-configuring controller could greatly increase the probability of mission success in a remote spacecraft, and motor driven actuators are frequently used, the application of hardware evolution to motor controller design is considered a good starting point for the development of a general self-configuring controller architecture.

2 Approach

The JPL developed Stand-Alone Board Level Evolvable (SABLE) System[3] is used for evolving analog control electronics. This system employs the JPL designed FPTA2. The FPTA2 contains 64 programmable cells on which an electronic design can be implemented by closing internal switches. The schematic diagram of one cell is given in the Appendix. Each cell has inputs and outputs connected to external pins or the outputs of neighboring cells. More detail on the FPTA2 architecture is found in [2]. A diagram of the experimental setup is shown in Figure 1. The main components of the system are a TI-6701 Digital Signal Processor (DSP), a 100kSa/sec 16-channel DAC and ADC and the FPTA2. There is a 32-bit digital I/O interface connecting the DSP to the FPTA2. The genetic algorithm (GA) running on the DSP follows a simple algorithm: download an individual, stimulate the circuit with a control signal, record the response, evaluate the response against the expected. This is repeated for each individual in the population and then crossover and mutation operators are performed on all but the elite percentage of individuals. The evolution is considered complete when a target fitness is reached, or when the fitness value plateaus and no further improvement is seen.

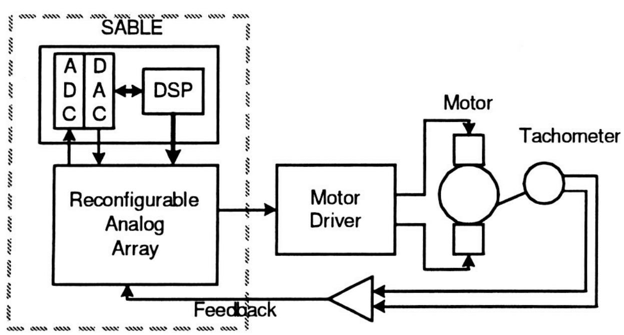


Figure 1. Configuration of the SABLE System and motor to be controlled

The motor used is a DC servo-motor with a tachometer mounted to the shaft of the motor. The motor driver is configured to accept motor current commands and requires a 17.5 volt power supply with the capability to produce 6 amps of current. A negative 17.5 volt supply with considerably lower current requirements is needed for the circuitry that translates FPTA2 output signals to the proper range for input to the driver. The tachometer feedback range is roughly [-4, +4] volts which corresponds to a motor shaft speed range of [-1300, +1300] RPM. Therefore, the tachometer feedback is biased to create a unipolar signal, then reduced in magnitude to the [0, 1.8] volt range the FPTA2 can accept.

For comparison with evolved controllers, a conventional analog controller is designed that can be directly substituted for the SABLE system. The response of the motor to the controller input from the SABLE system and the conventional controller is recorded and displayed using a digital storage oscilloscope. Data collected by the oscilloscope can be stored in a spreadsheet format and formatted for presentation in plots.

3 Conventional Analog Controller

3.1 Design

All closed-loop control systems require the calculation of an error measure, which is manipulated by the controller to produce a control input to the dynamic system being controlled, commonly referred to as the plant. The most widely used form of analog controller is a proportional-integral (PI) controller. This controller is frequently used to provide current control and speed control for a motor. The PI control law is given in Equation 1,

$$u(t) = K_P e(t) + \int \frac{1}{K_I} e(t) dt \qquad (1)$$

where $e(t)$ is the difference between the desired plant response and the actual plant response, K_P is the proportional gain, and K_I is the integral gain. In this control law, the proportional and integral terms are separate and added together to form the control input to the plant. The proportional gain is set to provide quick response to changes in the error, and the integral term is set to null out steady state error.

The FPTA2 is a unipolar device using voltages in the range of 0 to 1.8 volts. In order to directly compare a conventional analog controller design with evolved designs, the PI controller must be implemented as shown in Figure 2 using a supply voltage of 1.8V for the op-amps. This figure includes the circuitry needed to produce the error signal. Equation 2 gives the error voltage, V_e, given the desired response V_{SP}, or setpoint, and the measured motor speed V_{TACH}. The frequency domain transfer function for the voltage output, V_u, of the controller, given V_e, is shown in Equation 3,

$$\begin{aligned} V_e &= \frac{V_{SP}}{2} - \frac{V_{TACH}}{2} + 0.9V \qquad (2) \\ V_u &= (V_e - V_{bias2})(\frac{R_2}{R_1} + \frac{1}{sR_1C}) + V_e \qquad (3) \end{aligned}$$

where s is complex frequency in rad/sec, $\frac{R_2}{R_1}$ is the proportional gain and $\frac{1}{R_1C}$ corresponds to the integral gain. This conventional design requires four op-amps. Two are used to isolate voltage references V_{bias1} and V_{bias2} from the rest of the circuitry, thereby maintaining a steady bias voltage in each case. V_{bias2} must be adjusted to provide a plant response without a constant error bias. The values for R_1, R_2, and C are chosen to obtain the desired motor speed response.

3.2 Performance

The controller circuitry in Figure 2 is used to provide a baseline control response to compare with the responses obtained via evolution. The motor is run with no external torque load on the shaft. The controller is configured with $R_1 = 10K$ ohms, $R_2 = 200K$ ohms, and $C = 0.47uF$. V_{bias2} is set to 0.854 volts. Figure 3 illustrates the response obtained for V_{SP} consisting of a 2 Hz sinusoid with amplitude in the range of approximately 500 millivolts to 1.5 Volts, as well as for V_{SP} consisting of a 2 Hz square wave with the same magnitude. Statistical analysis of the error

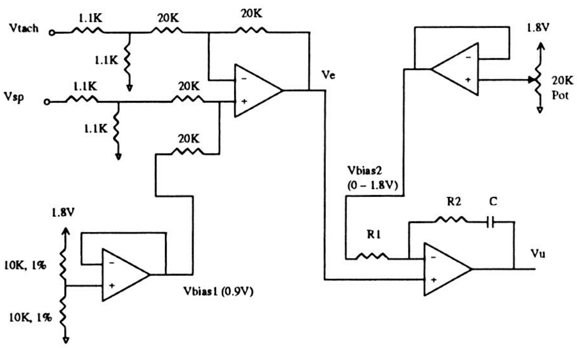


Figure 2. Unipolar analog PI controller with associated error signal calculation and voltage biasing

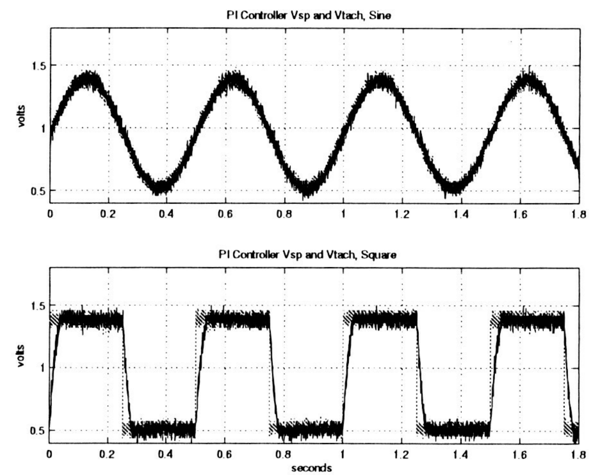


Figure 3. Response obtained using PI controller. Vsp is gray, Vtach is black.

between V_{TACH} and V_{SP} for sinusoidal V_{SP} is presented in Table 1 for comparison with the evolved controller responses. Table 2 gives the rise time and error statistics at steady state for the first full positive going transition in the square wave response. This is the equivalent of analyzing a step response. Note that in both cases V_{TACH} tracks V_{SP} very well. In the sinusoid case, there is no visible error between the two. For the square wave case, the only visible error is at the instant V_{SP} changes value. This is expected, because no practical servo-motor can follow instantaneous changes in speed. There is always some lag between the setpoint and response. After the transition, the PI controller does not overshoot the steady state setpoint value, and provides good regulation of motor shaft speed at the steady state values.

4 Evolved Controller

4.1 Baseline Design

Two cells within the FPTA2 are used in the evolution of the motor speed controllers. The primary cell is provided with the motor speed setpoint, V_{SP} , and the motor shaft feedback , V_{TACH}, as inputs, and it produces the controller output, V_u. An adjacent cell is used to provide support electronics for the first cell. The evolution uses a fitness function based on the error between V_{SP} and V_{TACH}. Lower fitness is better, because the goal is to minimize the error. The population is randomly generated, and then modified to ensure that, initially, Switches $S_{57}$and S_{53} (refer to the cell diagram in the Appendix) are closed to connect V_{SP} and V_{TACH} to the internal reconfigurable circuitry. This is done because the evolution will, in some cases, attempt to control the motor speed by using the setpoint signal only, resulting in an undesirable "controller" with poor response characteristics. By definition, a closed-loop controller must use both the command and a feedback signal, so the inclusion of this constraint is considered desirable. Many evolutions were run, and the frequency of the sinusoidal signal was varied, along with the population size and the fitness function. There were some experiments that failed to produce a desirable controller and some that produced very desirable responses, with a distribution of mediocre controllers in between . One of the evolved controllers is presented along with the response data for comparison to the PI controller. This controller is the best evolved controller obtained, at the time of writing.

For the evolution of this controller, the population size was 100 and a roughly 2.25 Hz sinusoidal signal was used for the setpoint. For a population of 100, the evaluation of each generation takes 45 seconds. The fitness function used is,

$$F = \alpha * \sum_{i=1}^{n} e_i^2 + \frac{\beta}{n} \sum_{i=1}^{n} |e_i| + M + N \tag{4}$$

$$M = \gamma_1 * |\max(V_{SP}) - \max(V_{TACH})| \tag{5}$$

$$N = \gamma_2 * |\min(V_{SP}) - \min(V_{TACH})| \tag{6}$$

where e_i is the error between V_{SP} and V_{TACH} at each voltage signal sample, n is the number of samples over one complete cycle of the sinusoidal input, and $\alpha, \beta, \gamma_1, \gamma_2$ are gains, or weights, for the different components of the fitness function. The evolution converged to a fitness of 42,897 at generation 442. The fitness values are large due to the small values of error that are always present in a physical system.

4.2 Performance

The response of the evolved controller is shown in Figure 4 for V_{SP} consisting of a 2 Hz sinusoid with amplitude in the range of approximately 500 millivolts to 1.5 Volts, as well as for V_{SP} consisting of a 2 Hz square wave with the same magnitude. This is the same input used to obtain controlled motor speed responses for the PI controller. The data presented in the plots was obtained by loading the previously evolved design on the FPTA2, and then providing V_{SP} via a function generator. The system response was recorded using a digital storage oscilloscope. In the sinusoidal case, the evolved controller is able to provide good peak to peak magnitude response and follows the sinusoidal curve well, but has a visible constant offset when compared to V_{SP} . The evolved controller provides a response to the square wave V_{SP}, which has a slightly longer rise time but provides good regulation of the speed at steady state with the addition of a constant offset . The statistical analysis of the evolved controller (EC) response to the sinusoidal V_{SP} is presented in Table 1. Note the increase in all the measures, with the mean error indicating a larger constant offset in the error response. Despite these increases, the controller response is reasonable, and exhibits the common voltage offset that frequently exists in analog control systems without trim potentiometers. The rise time and steady state error analysis for the first full positive going transition in the square wave response is given in Table 2. While there is an increase in rise time and in the error measures at steady state, when compared to those of the PI controller, the evolved controller can be considered to perform well. Note again that the increase in the mean error indicates a larger constant offset in the error response. In the PI controller, this error can be manually trimmed out via adjustment of V_{bias2}. The evolved controller has been given no such bias input, so some increase in steady state error should be expected. However, the baseline evolved controller is trimming this error, because other designs have a more significant error offset. Experiments with the evolved controller show that the "support" cell is providing the error biasing circuitry. This will be further illustrated in a later section.

It is notable that the evolved controller is providing a good response using a considerably different set of components than the PI controller. The evolved controller is using two adjacent cells in the FPTA to perform a similar function to four op-amps, a collection of 12 resistors and one capacitor. The FPTA switches have inherent resistance on the order of kilo-ohms, which can be exploited by evolution during the design. But the two cells can only be used to implement op-amp circuits similar to those in Figure 2 with the use of external resistors, capacitors and bias voltages. These external components are not provided.

The evolved design works on other pairs of cells in the FPTA2 equally well. The results are essentially indistinguishable, as illustrated in Figure 5. The error statistics for this case are included in Table 1 for comparison and are labeled as "moved EC". The differences are considered insignificant.

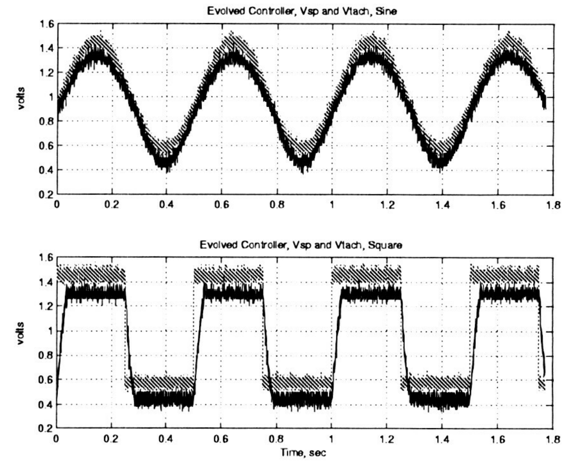


Figure 4. Response obtained using evolved controller, Vsp is gray, Vtach is black

Table 1. Error metrics for sinusoidal response

Control	Max Error	Mean Error	Std Dev Error	RMS Error
PI	0.16V	0.0028V	0.0430V	0.0431V
EC	0.28V	0.0875V	0.0520V	0.1018V
Moved EC	0.26V	0.0860V	0.0512V	0.1001V

Table 2. Response and error metrics for square wave. First full positive transition only

Control	Rise Time	Mean Error	Std Dev Error	RMS Error
PI	0.0358sec	0.0003V	0.0445V	0.0445V
EC	0.0382sec	0.1536V	0.0439V	0.1598V

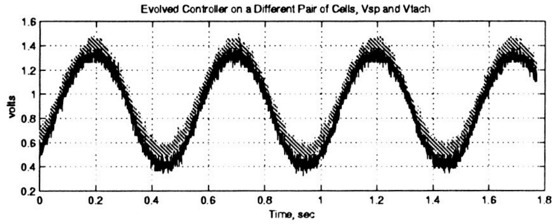


Figure 5. Response obtained using the evolved controller on a different pair of cells, Vsp is gray, Vtach is black

5 Reconfiguration for Fault Tolerance

Standard approaches to fault tolerance through hardware redundancy include passive methods using fault masking and active methods employing fault detection and reconfiguration. Systems requiring extremely high reliability may use a hybrid approach employing passive and active techniques. Passive methods rely on voting techniques and replicated hardware to mask the occurrence of faults. Active methods employ not only redundant hardware, but include fault detection, isolation and recovery through reconfiguration [17]. Hardware evolution offers an approach to fault tolerance that can use redundancy, if available, or reconfigure existing components. The ability to accommodate faults using hardware evolution was investigated by injecting "stuck-at" fault conditions into the cell configuration on the FPTA. Faults of this nature could result in a space environment due to radiation induced single event latchup (SEL) in the configuration circuitry on the FPTA.

In the previous section, results are presented that show the controller operates equally well on cells other than those on which it was evolved. This leads to the idea that fault tolerance can be provided by reserving a pool of unused cells, and drawing on these in the event of a controller fault. Controller faults could be identified and accommodated by measuring fitness on a periodic basis. When the fitness exceeds specified bounds, the configuration for the design is loaded again to accommodate a transient fault in the configuration that can be corrected by refreshing the configuration bits. If this does not improve the response, the design is moved to another set of known good cells to determine if the fault is in the original cells. When the pool of reserve cells has been exhausted, hardware evolution can be employed to reconfigure an evolved controller in the available cells to accommodate faults by recovering some, if not all, of its original capability. Depending on the damage that has caused the fault, hardware evolution may be able to make use of damaged components, if these components

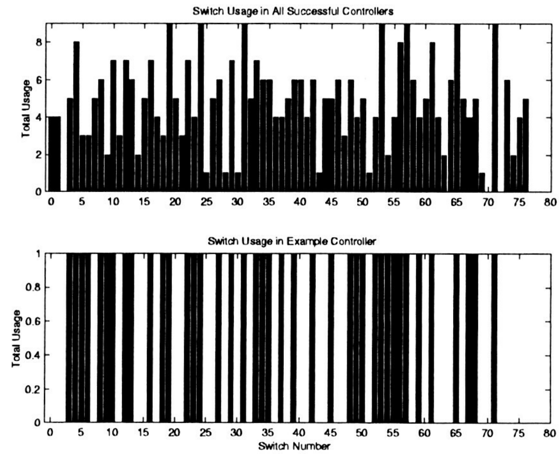


Figure 6. Histograms of switch usage in the primary cell in multiple controller designs (upper plot) and the baseline evolved controller (lower plot)

retain partial, or degraded, functionality

For these experiments, it is assumed that the motor being controlled can be decoupled from its load during evolution. This can be achieved using a clutch, which adds complexity, but would be needed in this case. For continued operation of the motor driven system during evolution, a redundant controller and motor must be switched in, since the output of the controller does not always provide desirable results during the evaluation of individuals in the population. It is recognized that the added mechanical complexity may be undesirable in some applications. However, actuators driven by redundant motors are frequently applied in aerospace applications.

In the evolved design, some switches are considered critical, in that opening these switches will either damage the design beyond repair or, create a situation in which only partial capability can be recovered. Some of these switches can be identified by simply observing their usage across multiple evolved designs. Figure 6 is a histogram showing switch usage in the primary cell across nine successful controller designs in the upper plot. Refer to the cell diagram in the Appendix for the location of these switches. These nine include the baseline controller in the previous section, and the lower plot shows switch usage in the baseline controller only. Note that there are seven switches that are used in all the designs. Some of these switches represent clearly critical switches that will result in the complete failure of the design on a given set of cells. Switches 53 and 57 are examples of such switches. If either of these switches is opened, either V_{SP} or V_{TACH} will be isolated from the cells. However, other switches in this group are not as critical. For instance opening switch 65 has no effect on the performance of the baseline controller. Opening switches 71 and 19 cause the output of the baseline design to flat line, which implies severe damage. Opening switches 24 and 31 causes the motor speed to switch back and forth between the positive and negative speed limits in response to the sinusoidal command input. This implies a possibility for recovery, since the controller is still reacting to its inputs. From the plot, it appears that there are several other switches that may be important because of their use in six, seven or eight of the nine designs. One switch of interest in this category is switch 56, which connects output of the support cell to the input of the primary cell. Because this switch is not used in one of the successful designs, it is reasonable to assume evolution can find a new configuration that will recover functionality.

To illustrate the capability of hardware evolution to reconfigure the controller design to recover performance in the presence of faults, the SABLE software is configured to force the bit controlling a switch to open, or close, the switch of interest, regardless of the bit value in the individual configuration. A population approach to fault tolerance is used by loading the population obtained from the evolution resulting in the baseline controller, and then restarting the evolution. Experiments with some of the controllers have indicated that there is either an existing design in the population that recovers performance, or that evolution can find a way around a fault induced by opening some of the switches. In the following subsections, the results of three selected experiments in recovering performance after an induced fault in the design will be presented.

5.1 Fault Experiment A

In this experiment, switch 56 is opened, which takes away the connection between the output of the support cell from the primary cell. The upper plot in Figure 7 shows the resulting loss of performance when this switch is opened. The support cell is clearly seen to provide a biasing function for the controller input to the motor driver, and now that this function is lost the motor speed has a larger constant offset and saturates at a value of the motor speed in the "negative" direction of rotation.

The recovered response is shown in the lower plot of Figure 7. The controller used to achieve this response was obtained at generation 58 with a fitness of 144,037. Continuing the evolution for another 108 generations did not result in a controller with better fitness, and the experiment was halted. In this case evolution was able to improve the response for the upper half of the sinusoid and the maximum peak of V_{SP} , but undershoots the minimum peak. However, this response recovers the performance to a degree that will allow the motor to continue operating with a little degradation, and some improvement for the upper portion of the sinusoidal speed response.

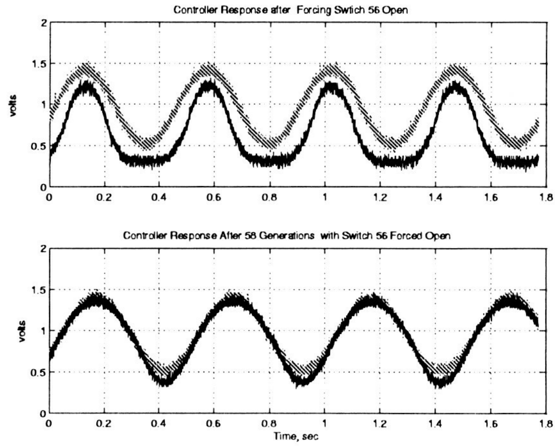


Figure 7. Response after inducing a fault by opening switch 56, and recovered response after evolution. Vsp is gray and Vtach is black

5.2 Fault Experiment B

For this case a switch 31 was chosen, because it is one of the seven switches used by all the successful designs. Opening this switch causes a severe degradation in the motor speed response as shown in the upper plot of Figure 8. The recovered response is shown in the lower plot of the figure. The controller used to achieve this response was obtained at generation 495 with a fitness of 633,531. Note that for this case with a more severe induced fault, the evolution took a great deal longer time to recover a reasonable level of functionality. Also the recovered functionality does not follow the sinusoidal V_{SP} as well now. The motor speed response is more similar to a triangle wave. But this recovered performance is clearly preferable to that of the damaged performance, and will allow the motor to be operated with somewhat degraded speed control.

5.3 Fault Experiment C

In this case, two switches are opened to determine the capabilities of evolution to handle multiple faults. The two switches selected are switch 56 and 29. Switch 29 is used in seven of the controller designs. The performance as a

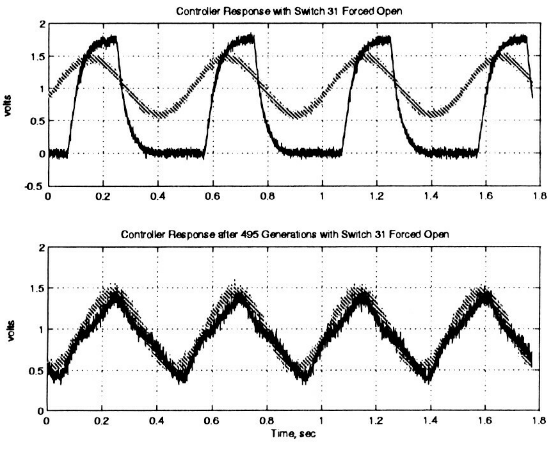


Figure 8. Response after inducing a fault by opening switch 31, and recovered response after evolution. Vsp is gray and Vtach is black

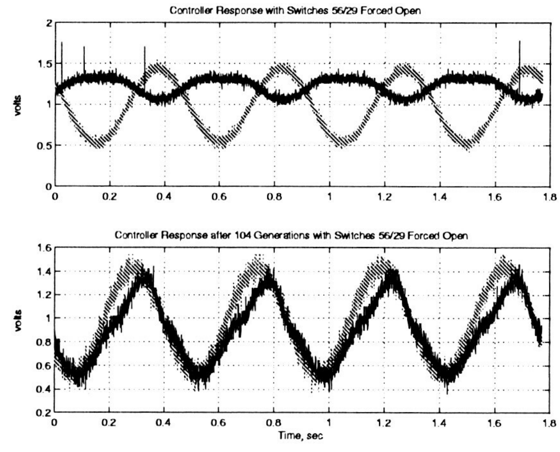


Figure 9. Response after inducing a fault by opening switches 56 and 29, and recovered response after evolution. Vsp is gray and Vtach is black

result of the damage caused to the design is shown in the upper plot of Figure 9. The design that produced the recovered response in the lower plot was obtained after 104 generations and has a fitness of 824,407. The recovered performance is clearly degraded in its overall ability to follow V_{SP} , but still provides a reasonable control of motor speed, in light of the response prior to the evolution.

6 Summary

The results presented show that it is possible to use the FPTA2 to evolve simple analog closed-loop controllers. The use of two cells to produce a controller that provides good response in comparison with a conventional controller shows that hardware evolution is able to create a compact design that still performs as required, while using less transistors than the conventional design, and no external components. Recall that one cell can be used to implement an op-amp design on the FPTA2. While a programmable device has programming overhead that fixed discrete electronic and integrated circuit components do not, this overhead is typically neglected when comparing the design on the programmable device to a design using fixed components. The programming overhead is indirect, and is not a functional component of the design. As such, the cell diagram in the Appendix shows that each cell contains 15 transistors available for use as functional components in the design. Switches have a finite resistance, and therefore functionally appear as passive components in a cell. The *simplified* diagram in the data sheets for many op-amps show the utilization of 30, or more, transistors in their design.

In order to produce self-configuring controllers that can rapidly converge to provide desired performance, more work is needed to speed up the evolution and guide it to the best response. The per generation evaluation time of 45 or more seconds is a bottleneck to achieving this goal. Further, the time constants of a real servo-motor may make it impossible to achieve more rapid evaluation times. Most servo-motor driven actuators cannot respond to inputs with frequency content of more than a few tens of Hertz, without attenuation in the response. Alternative methods of guiding the evolution or novel controller structures are required.

Additionally, it was shown that evolution can be used to reconfigure the design to recover performance after induced faults. The approach used allows a reconfigured controller to make use of cells in which there are damaged switches. This may produce degraded performance depending on the switches involved, and, in this case, should be used only when the controller cannot be moved to other cells. Future work will include experiments with controller reconfiguration in response to faults in the motor and its driver and to changes in the operational environment.

A key to improving upon this work and evolving more complex controllers is a good understanding of the circuits that have been evolved. Evolution has been shown to make use of parasitic effects and to use standard components in novel, and often difficult to understand, ways. Gaining this understanding may prove to be useful in developing techniques for guiding the evolution towards rapid convergence.

7 Acknowledgements

The authors would like to thank Jim Steincamp and Adrian Stoica for establishing the initial contact between Marshall Space Flight Center and the Jet Propulsion Laboratory leading to the collaboration for this work. The Marshall team appreciates JPL making available their FPTA2 chips and SABLE system design for conducting the experiments. Jim Steincamp's continued support and helpful insights into the application of genetic algorithms have been a significant contribution to this effort.

References

[1] Lohn, J. D. and Columbano, S. P., *A Circuit Representation Technique for Automated Circuit Design*, IEEE Trans. on Evolutionary Computation, Vol. 3, No. 3, September 1999.

[2] Stoica, A., Zebulum, R., Keymeulen, D., *Progress and Challenges in Building Evolvable Devices*, Evolvable Hardware, Proceedings of the third NASA/DoD Workshop on, July 2001, pp 33 – 35.

[3] Ferguson, M. I., Zebulum, R., Keymeulen, D. and Stoica, A., *An Evolvable Hardware Platform Based on DSP and FPTA*, Late Breaking Papers at the Genetic and Evolutionary Computation Conf. (GECCO-2002), July 2002, pp. 145-152.

[4] Stoica, A., Zebulum, R., Ferguson, M. I., Keymeulen, D. and Duong V., *Evolving Circuits in Seconds: Experiments with a Stand-Alone Board Level Evolvable System*, 2002 NASA/DoD Conf. on Evolvable Hardware, July 2002, pp. 67-74.

[5] Langeheine, J., Meier, K., Schemmel, J., *Intrinsic Evolution of Quasi DC solutions for Transistor Level Analog Electronic Circuits Using a CMOS FTPA Chip,* 2002 NASA/DoD Conf. on Evolvable Hardware, July 2002, pp. 75-84.

[6] Flockton, S. J. and Sheehan, K., "Evolvable Hardware Systems Using Programmable Analogue Devices", Evolvable Hardware Systems (Digest No. 1998/233), IEE Half-day Colloquium on , 1998 ,Page(s): 5/1 -5/6.

[7] Ozsvald, Ian, "Short-Circuit the Design Process: Evolutionary Algorithms for Circuit Design using Reconfigurable Analogue Hardware", Master's Thesis, University of Sussex, September, 1998.

[8] Koza,J. R., Keane, M. A., Yu, J., Mydlowec, W. and Bennet, F., *Automatic Synthesis of Both the Control Law and Parameters for a Controller for a Three-lag plant with Five-Second delay using Genetic Programming and Simulation Techniques*, American Control Conf., June 2000.

[9] Keane, M. A., Koza, J. R., and Streeter, M.J., *Automatic Synthesis Using Genetic Programming of an Improved General-Purpose Controller for Industrially Representative Plants*, 2002 NASA/DoD Conf. on Evolvable Hardware, July 2002, pp. 67-74.

[10] Zebulum, R. S., Pacheco, M. A., Vellasco, M., Sinohara, H. T., *Evolvable Hardware: On the Automatic Synthesis of Analog Control Systems*, 2000 IEEE Aerospace Conference Proc., March 2000, pp 451-463.

[11] Canham, R. O. and Tyrrell, A. M., *Evolved Fault Tolerance in Evolvable Hardware*, Proc. of the Congress on Evolutionary Computation 2002 (CEC2002), Honolulu, Hawaii, May 2002.

[12] Lohn, Jason, Larchev, Greg, and DeMara, Ronald, *A Genetic Representation for Evolutionary Fault Recovery in Virtex FPGAs,* Proc. of the 5th International Conf. on Evolvable Systems, Trondheim, Norway, March 2003.

[13] Keymeulen, Didier, Zebulum, Ricard Salem, Jin, Yili, and Stoica, Adrian, *Fault-tolerant Evolvable Hardware Using Field Programmable Transistor Arrays,* IEEE Trans. on Reliability, Vol. 29, No.3, September 2000, pp 305-316.

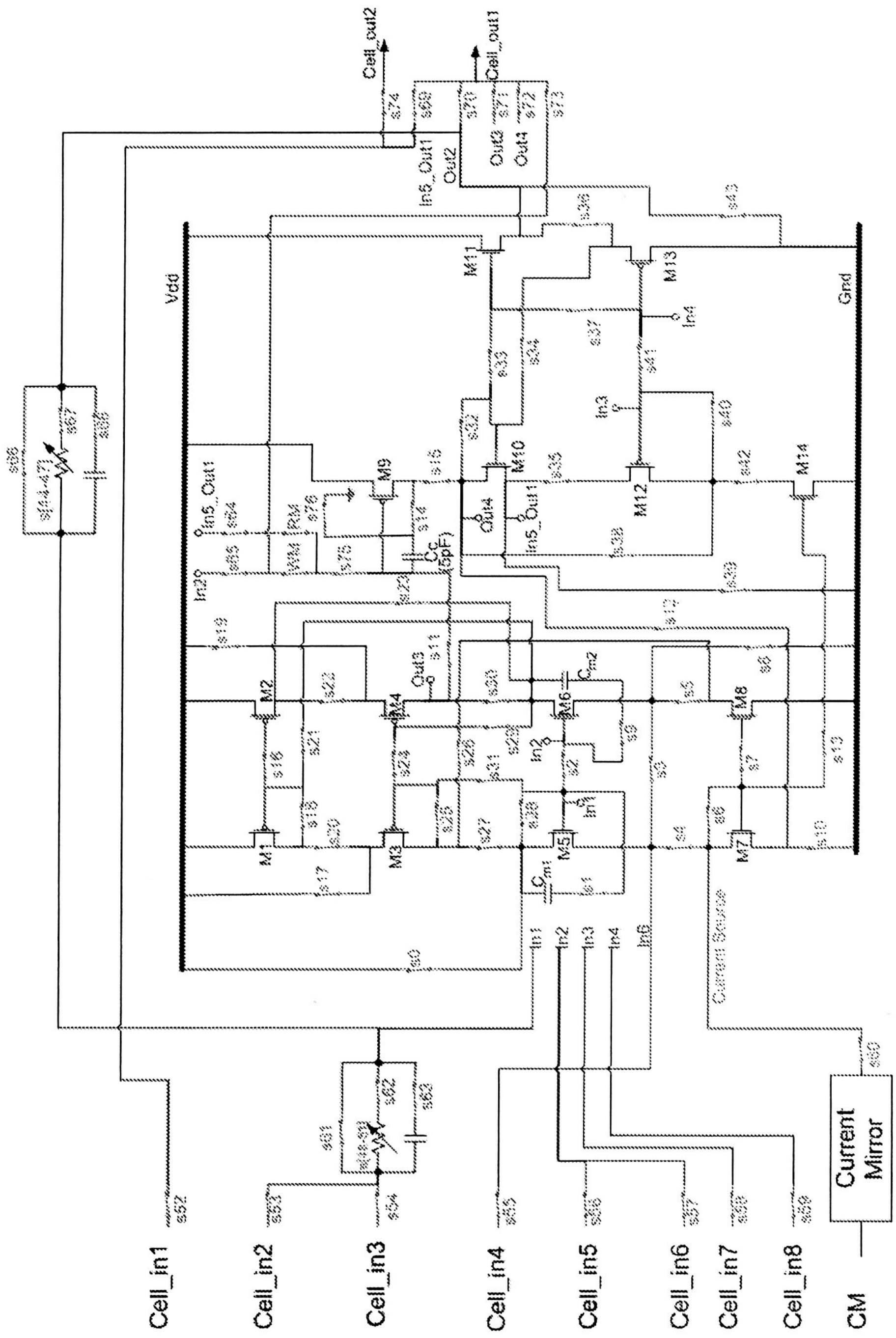
[14] Raimondi, G. M., et. al., *Large Electromechanical Actuation Systems for Flight Control Surfaces*, IEE Colloquium on All Electronic Aircraft, 1998.

[15] Jensen, S.C., Jenney, G. D., Raymond, B., Dawson, D., *Flight Test Experience with an Electromechanicl Actuator on the F-18 Systems Research Aircraft,* Proceedings of the 19th Digital Avionics System Conference, Volume 1, 2000.

[16] Byrd, V. T., Parker, J. K, *Further Consideration of an Electromechanical Thrust Vector Control Actuator Experiencing Large Magnitude Collinear Transient Forces*, Proc. of the 29th Southeastern Symposium on System Theory, March 1997, pp 338-342.

[17] Johnson, Barry W., *Design and Analysis of Fault-Tolerant Digital Systems,* Addison-Wesley Publishing Company, Reading Massachusetts, 1989.

Appendix: FPTA2 Cell Diagram



Intrinsic Hardware Evolution for the Design and Reconfiguration of Analog Speed Controllers for a DC Motor

D. A. Gwaltney, NASA Marshall Space Flight Center
M. I. Ferguson, Jet Propulsion Laboratory

2003 NASA/DoD Conference on Evolvable Hardware
July 9-11, 2003

Outline

- ⇨ Hardware Evolution of Electronic Circuitry
- ⇨ Fault Tolerance
- ⇨ Evolution of Controllers
- ⇨ Motivation for This Work
- ⇨ Approach
- ⇨ Baseline Analog Controller Design
- ⇨ Evolution of Analog Controllers
- ⇨ Experimental Results
- ⇨ Summary

Hardware Evolution of Electronic Circuitry

- ⇨ Two “basic” approaches
  - → Intrinsic
    - Commercial FPGAs and FPAA
    - Field Programmable Transistor Arrays (FPTA)
    - Multiplexed Components
  - → Extrinsic
    - SPICE Simulation for circuitry
    - Device specific simulations (Virtex DS)
- ⇨ Hybrid simulation/hardware
  - Compute intensive analysis in software combined with configuration of reconfigurable hardware
    - Evolution in simulation, then implemented
    - Evolution in hardware controlling simulated system

Fault Tolerance

- ⇨ Faults can be internal or external to evolved circuitry
  - → Most published results refer to internal faults
- ⇨ Approaches to evolving FT Circuitry
  - → Population approach
    - Select appropriate individual from existing population
    - Seed population with known designs
    - Continue evolution using existing/seeded population
    - Can be used for repair or reconfiguration
  - → Fault injection during evolution
    - May require a priori knowledge of possible faults
    - Random faults can be used for internal faults in large FPGA-based designs
    - Useful for design of analog circuitry with a relatively small number of components

Evolution of Closed-loop Controllers

- ⇨ Many evolved controllers suitable only for implementation in software
  - → Control law synthesis and tuning
    - Conventional transfer function or state space designs
    - Combination of Neural Networks and/or Fuzzy Logic and Evolution
  - → Evolution performed using simulation of controlled dynamics and controller
- ⇨ Analog controller evolution using plant simulation and electronics simulation using SPICE
- ⇨ Robot control using extrinsic and hybrid approach

Motivation for This Work

- ⇨ Development of an autonomous, self-configurable controller for use on a remotely located platform
  - → Adaptation of controller to environmental changes that would otherwise degrade performance
    - Temperature extremes
    - Ionizing radiation
    - Variation in the controlled dynamic system
  - → Reconfiguration of controller structure
    - Use of degraded components: electronic and mechanical
    - Exclusion of failed components: sensor failures
  - → Response to changing mission requirements
- ⇨ Applicable to platforms for exploration of space or extreme terrestrial environments

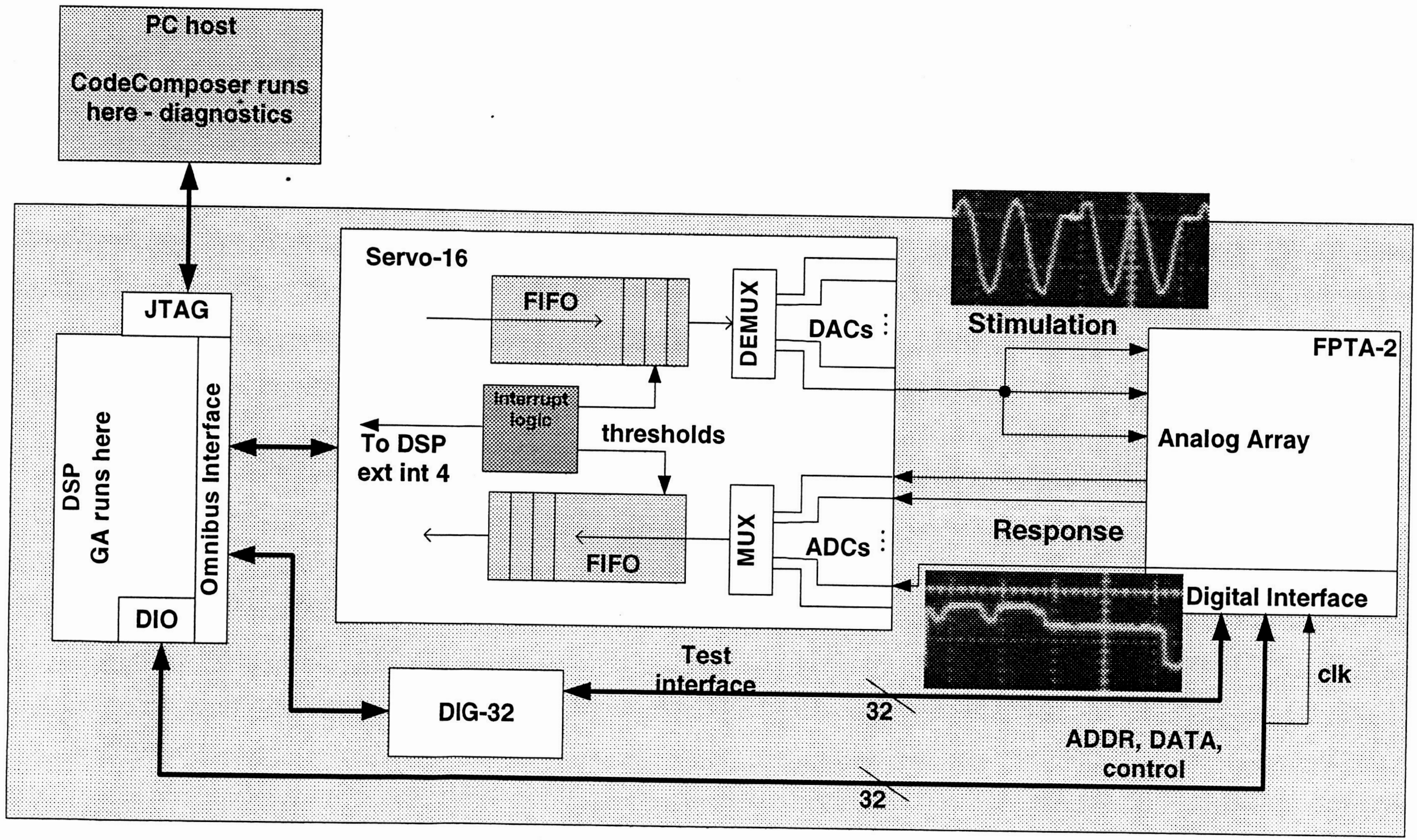
Motivation for This Work

- Motor driven systems are ubiquitous in commercial, industrial, military and aerospace applications
- By-wire technology relies extensively on electromechanical actuators
  - Aircraft, fly-by-wire
  - Automobiles, drive-by-wire
  - Recent space transportation vehicle concepts
- Availability of programmable analog or mixed-signal devices makes the implementation of analog control loops attractive at the actuator and subsystem level
  - Can be as easily modified as software

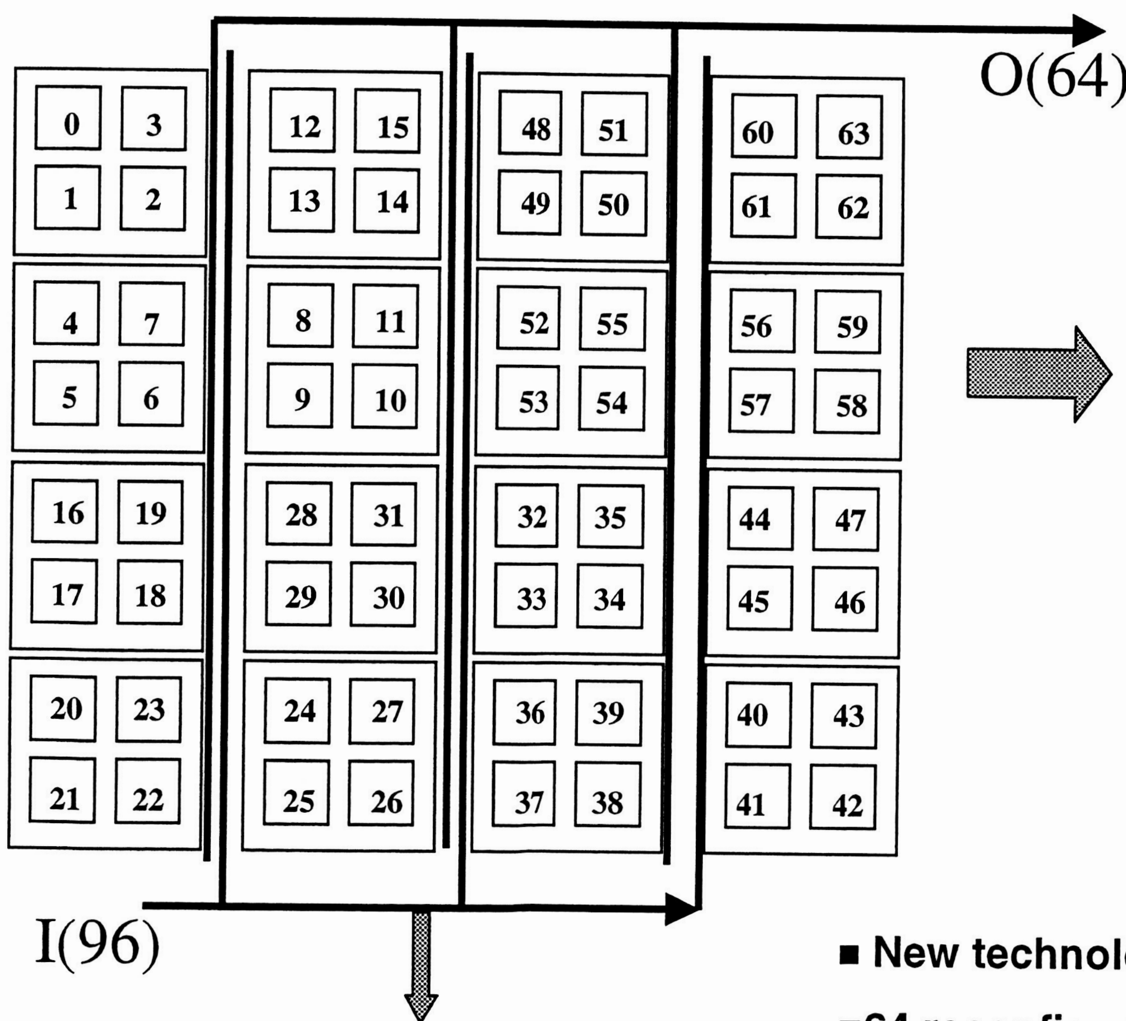
Approach

- ⇨ Use of JPL developed Stand Alone Board Level Evolvable (SABLE) System as platform for controller evolution
  - → Employs the JPL second generation Field Programmable Transistor Array (FPTA2)
  - → GA implemented in firmware and executed on a Digital Signal Processor (DSP) single board computer (SBC)
  - → FPTA2 configuration through digital interface on SBC
  - → 16 bit DAC provides stimulus of the FPTA2
  - → 16-bit , multi-channel ADC module captures FPTA2 output and other analog signals for fitness evaluation

SABLES

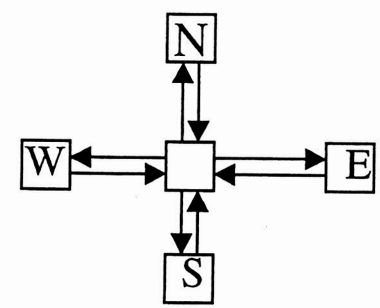

PC host
CodeComposer runs here - diagnostics
JTAG
DSP
GA runs here
DIO
Omnibus Interface
Servo-16
FIFO
DEMUX
DACs
Interrupt logic
thresholds
To DSP ext int 4
FIFO
MUX
ADCs
DIG-32
Test interface
32
32
Stimulation
Response
FPTA-2
Analog Array
Digital Interface
clk
ADDR, DATA, control

JPL FPTA2 Cell Layout



96 analog inputs
64 analog outputs

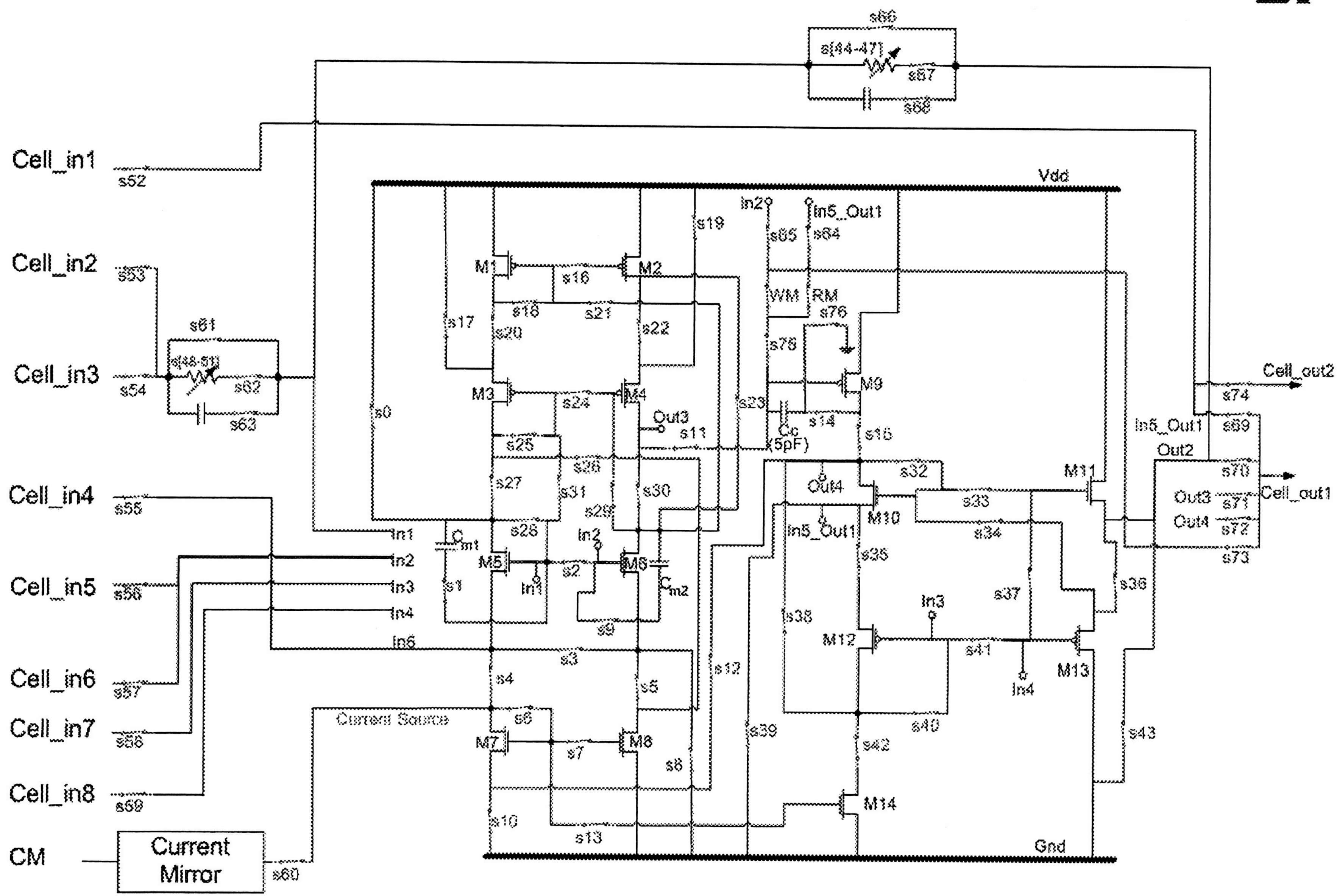
Interconnection between cells



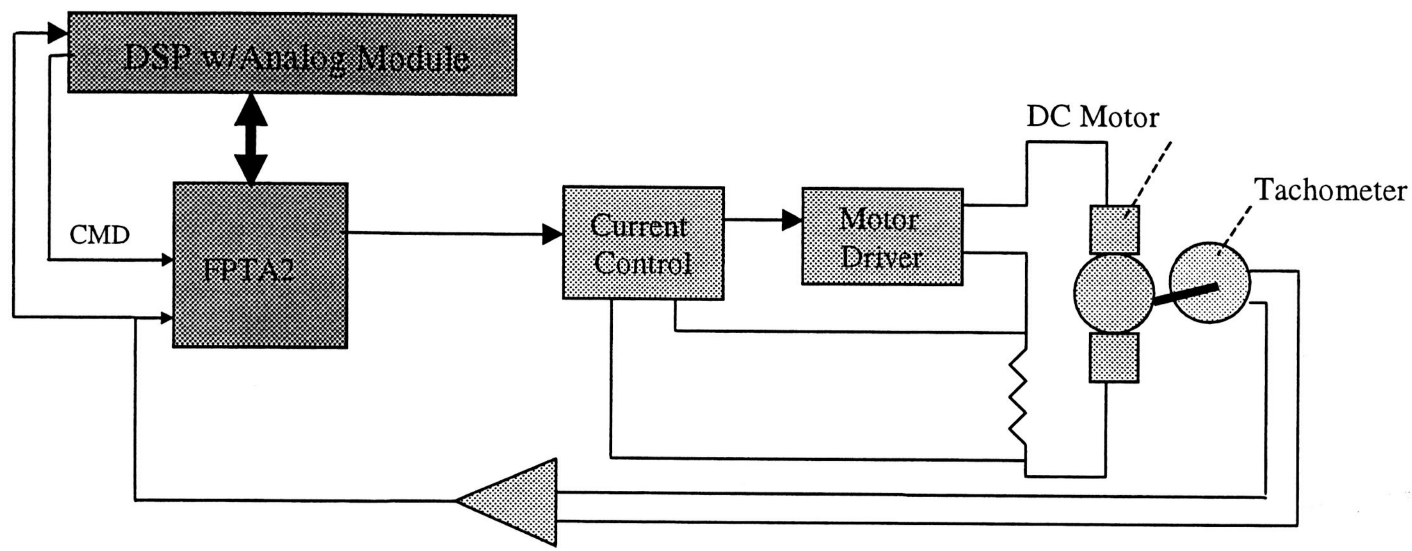
- New technology: TSMC 0.18 – 1.8v;
- 64 reconfigurable cells;

Marshall Space Flight Center
Jet Propulsion Laboratory

JPL FPTA2 Cell Schematic



Evolvable Controller Configuration

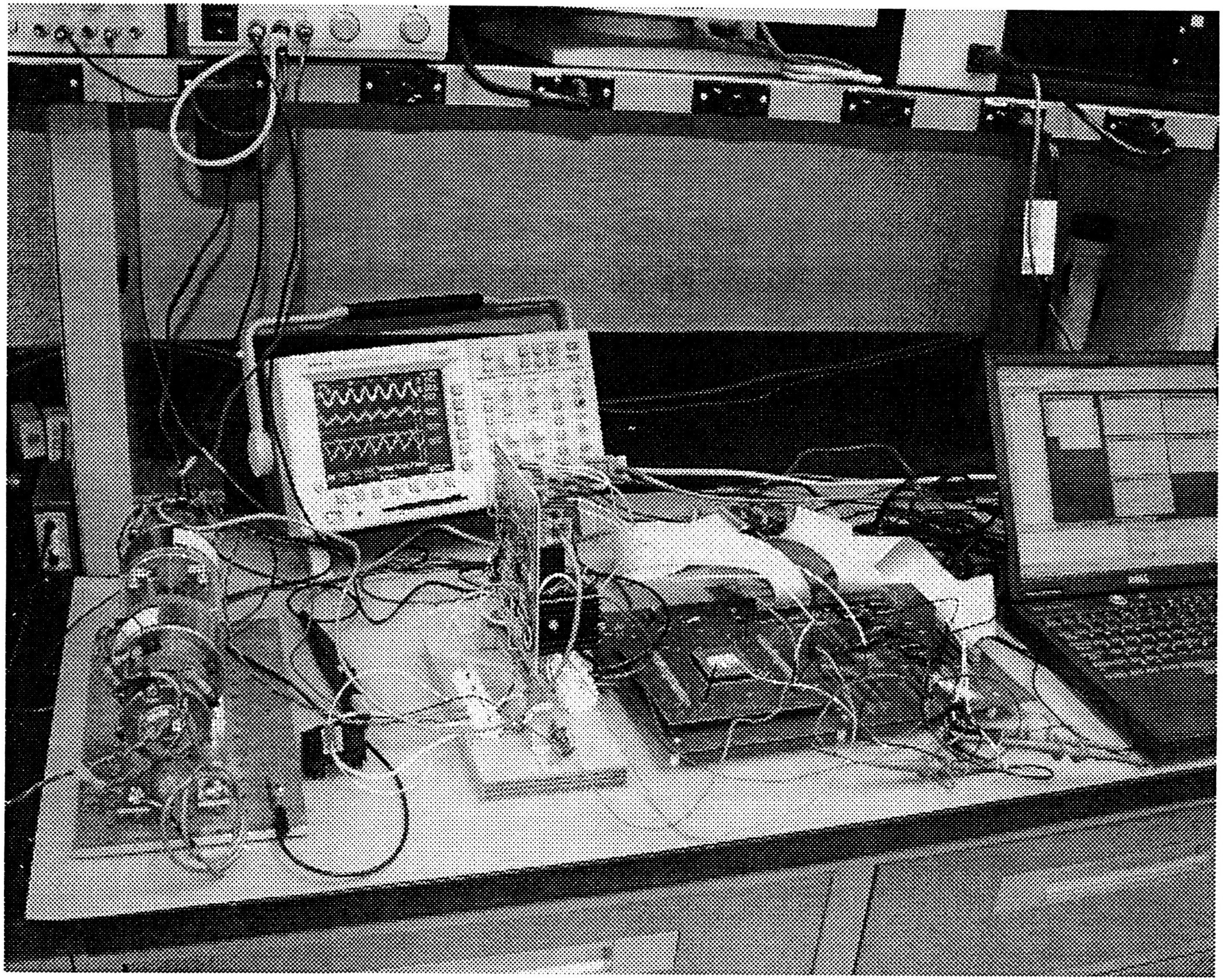


Error = CMD - Feedback

Diagram of the experimental configuration for hardware evolution of analog motor speed controllers

Evolvable Controller Configuration

Marshall Space Flight Center
Jet Propulsion Laboratory



Baseline Analog Controller Design

- ⇨ Most widely used form of analog controller is the Proportional-Integral (PI) Controller
  - → Used for motor current control and speed control

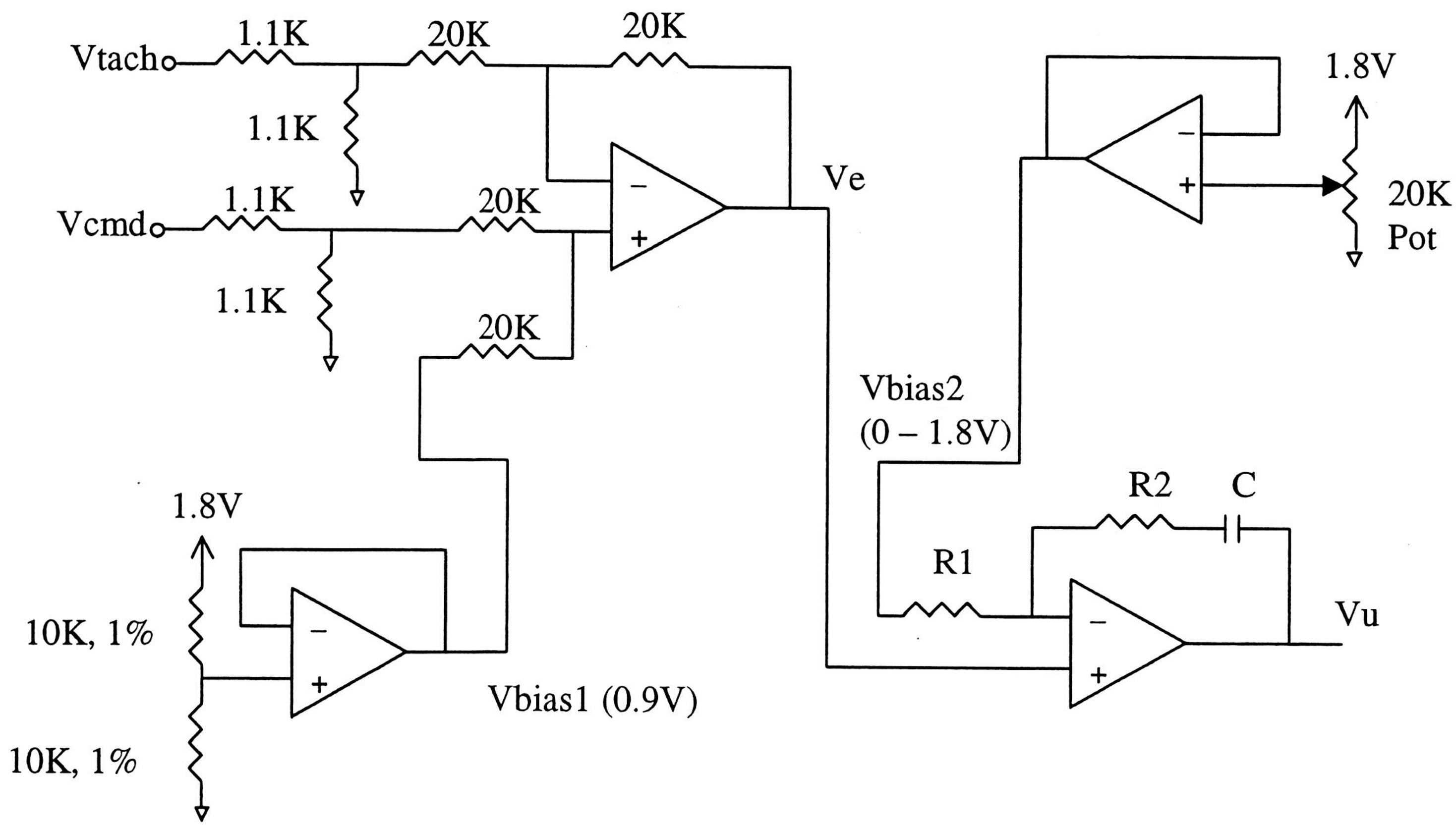
PI control law is represented by

$$u(t) = K_P e(t) + \frac{1}{(K_I)} \int e(t) dt$$

Where *e(t)* is error between desired plant response and measured plant response,
K_P is proportional gain,
K_I is integral gain or time constant

Baseline Analog PI Controller

PI Controller for 1.8V supply and 1.8V unipolar signal range



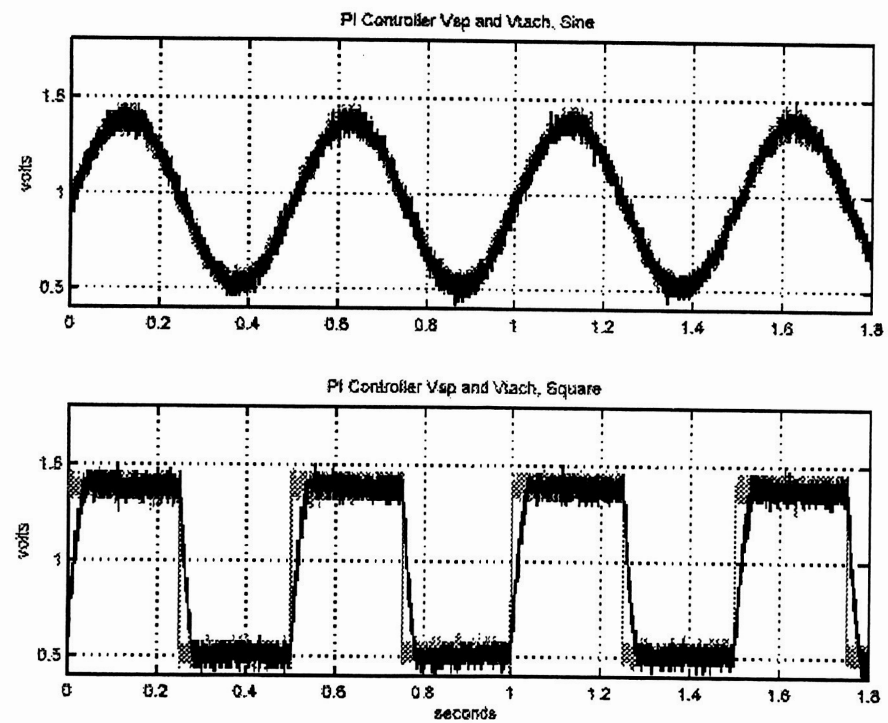
Vu = (Ve - Vbias2)(R2/R1 + 1/R1Cs) + Ve

Ve = Vcmd/2 +0.9 – Vtach/2

R1 = 10K, R2 =200K, C=0.47uF, Vbias2 = 0.865

Vcmd and Vtach, Ve and Vu are 0 – 1.8V signals

Response Using PI Controller



Response with no offset between command and tachometer feedback is achieved by adjusting Vbias2

Motor speed response obtained using PI controller. Vsp is gray, Vtach is black

Evolution of Analog Controllers

- Two FPTA cells used
  - Cell 1 provided with motor speed command and tachometer feedback. Responsible for producing current command to motor driver.
  - Cell 0 used to provide support electronics for cell 1
- Evolutionary Algorithm
  - Standard GA
  - Population initially seeded with randomly generated FPTA2 configurations
  - Population constrained to force the cell 1 switches closed that connect the speed command and tachometer feedback to the cell
    - All closed-loop controllers must use a command and feedback to produce an error signal
- DAC module generates sinusoid for motor speed command
- ADC module simultaneously records command and feedback signals for fitness evaluation

Evolution of Analog Controllers

- Many Controllers were evolved using varying fitness functions and conditions
- Experimental results for one evolved controller will be presented. The fitness function and conditions for the evolution is shown below

$$F = \alpha * \sum_{i=1}^{n} e_i^2 + \left| \frac{\beta}{n} \sum_{i=1}^{n} e_i \right| + M + N$$

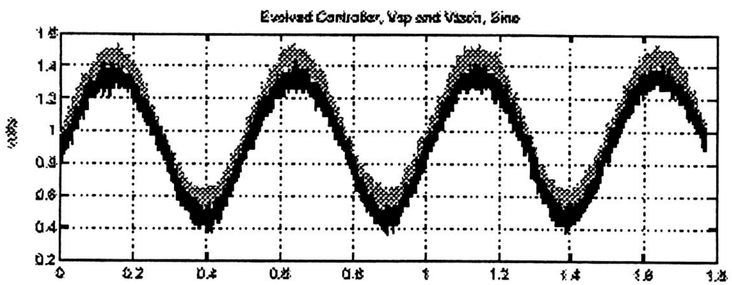
$$M = \gamma_1 * \left| \max(V_{SP}) - \max(V_{TACH}) \right|$$

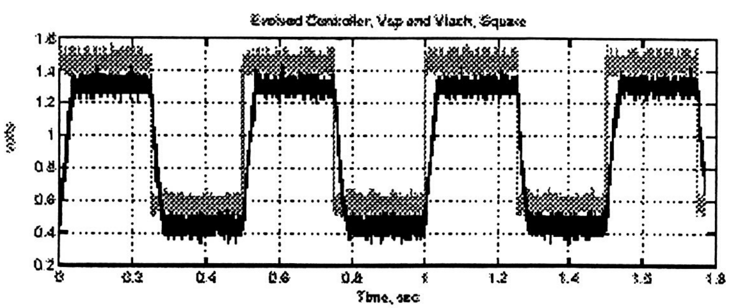
$$N = \gamma_2 * \left| \min(V_{SP}) - \min(V_{TACH}) \right|$$

•Goal is to minimize error and therefore the value of the fitness
•Forced closure of switches connecting speed command and feedback signal to cell 1
•100 individuals in the population
•2 Hz sinusoid for speed command

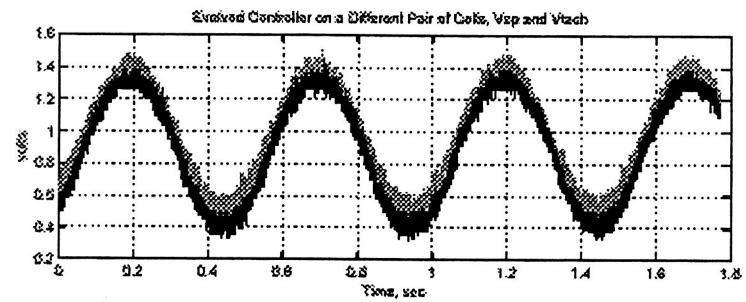
Experimental Results

Evolved Controller





Motor speed response obtained using evolved controller in original cells. Vsp is gray, Vtach is black



Motor speed response obtained using evolved controller in a different pair of cells. Vsp is gray, Vtach is black

Differences in response are considered insignificant.

Experimental Results

The evolved controller response is compared to the baseline response obtained using the PI controller in the tables below

Table 1. Error metrics for sinusoidal response

Control	Max Error	Mean Error	Std Dev Error	RMS Error
PI	0.16V	0.0028V	0.0430V	0.0431V
EC	0.28V	0.0875V	0.0520V	0.1018V
Moved EC	0.26V	0.0860V	0.0512V	0.1001V

Table 2. Response and error metrics for square wave. First full positive transition only

Control	Rise Time	Mean Error	Std Dev Error	RMS Error
PI	0.0358sec	0.0003V	0.0445V	0.0445V
EC	0.0382sec	0.1536V	0.0439V	0.1598V

- The increase in mean error indicates higher constant offset error in the response. In PI controller this offset is removed via adjustment of Vbias2. FPTA is given no such bias input.
- Experiments show the cell 0 is providing a biasing action for cell 1

Experimental Results

- The evolved controller provides a good response using two adjacent cells in the FPTA to perform a similar function to four op-amps, a collection of 12 resistors and one capacitor.
  - The FPTA switches have inherent resistance on the order of KOhms
  - Each cell has 14 transistors to be used as functional components
  - Cells can be used to implement op-amps by using external passive components and bias voltages
    - No external components or bias provided
- The capability of the controller to respond to inputs of varying amplitude and frequency and varying load was demonstrated, but the results are not presented here

Reconfiguration for Fault Tolerance

- ⇨ Reconfiguration for fault tolerance via evolution
  - → Use after pool of reserve cells is exhausted
  - → Assuming capability to remove motor from driven load through a clutch
  - → Continued operation of controlled component requires a redundant controller/motor configuration.
    - Dual redundant drives are not uncommon in critical actuator components for aerospace applications
- ⇨ For remote spacecraft, this is a near end-of-life approach to extend functional operation, or a last resort in the event of a catastrophic failure.

Reconfiguration for Fault Tolerance

Marshall Space Flight Center
Jet Propulsion Laboratory

- ⇨ Investigated controller internal faults simulated by forcing open switches used in the primary cell
- ⇨ Population approach to FT
- ⇨ Selected switches for faults by considering usage in successful designs
  - → Seven switches used in all successful designs
    - • S53 & S57 required to connect external signals
    - • S71 & S19 cause output to flat line
    - • S24 & s31 cause output switch between limits
    - • S65 has no effect
  - → Other switches used less frequently still critical to controllers
    - • S56 connects the two cells

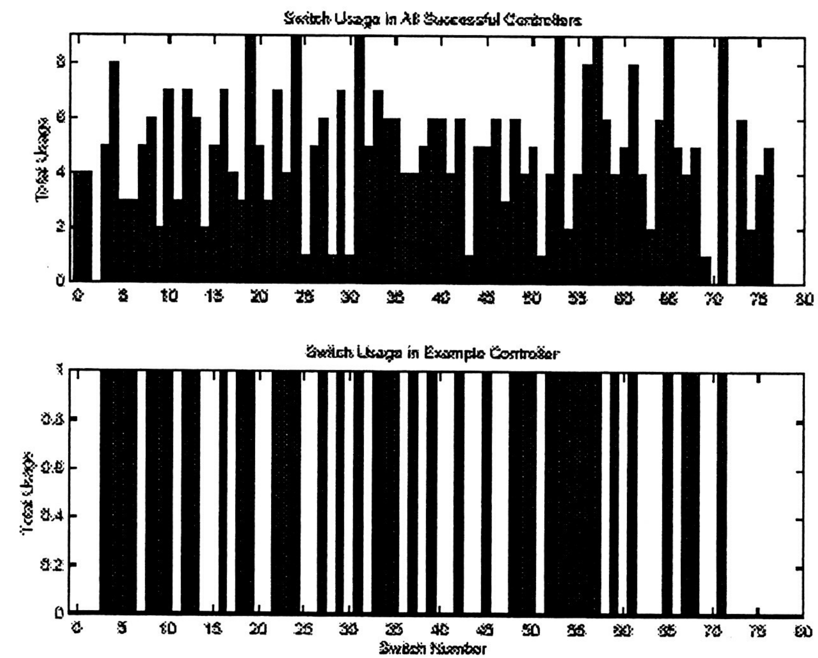


Figure 6. Histograms of switch usage in the primary cell in multiple controller designs (upper plot) and the baseline evolved controller (lower plot)

Marshall Space Flight Center
Jet Propulsion Laboratory

Reconfiguration for Fault Tolerance

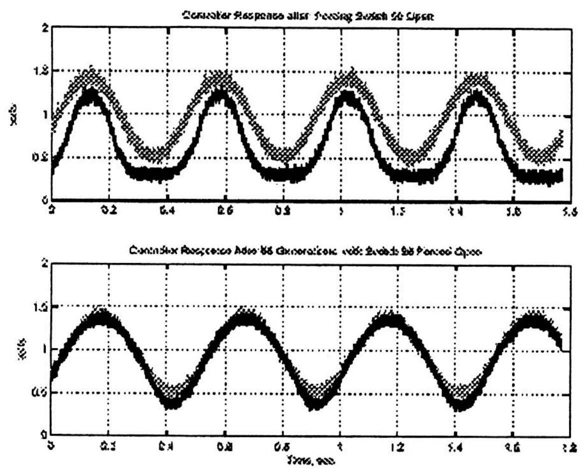


Figure 7. Response after inducing a fault by opening switch 56, and recovered response after evolution. Vsp is gray and Vtach is black

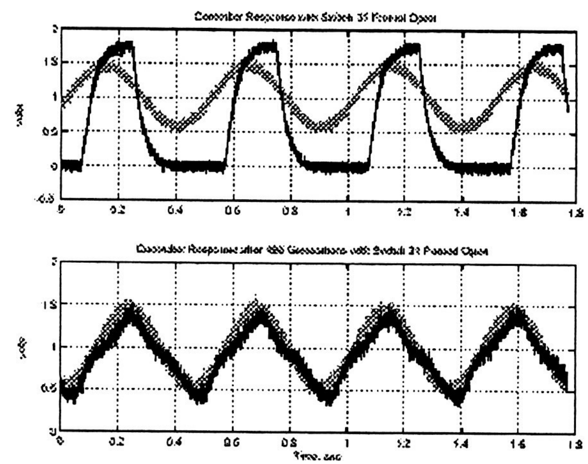


Figure 8. Response after inducing a fault by opening switch 31, and recovered response after evolution. Vsp is gray and Vtach is black

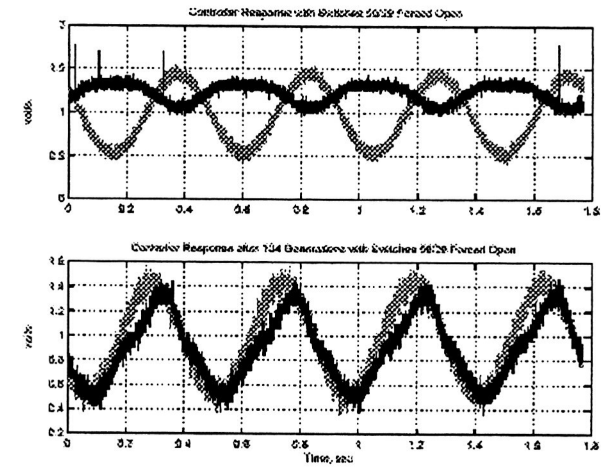


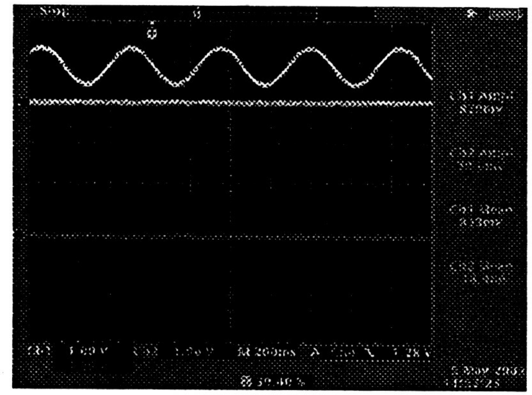
Figure 9. Response after inducing a fault by opening switches 56 and 29, and recovered response after evolution. Vsp is gray and Vtach is black

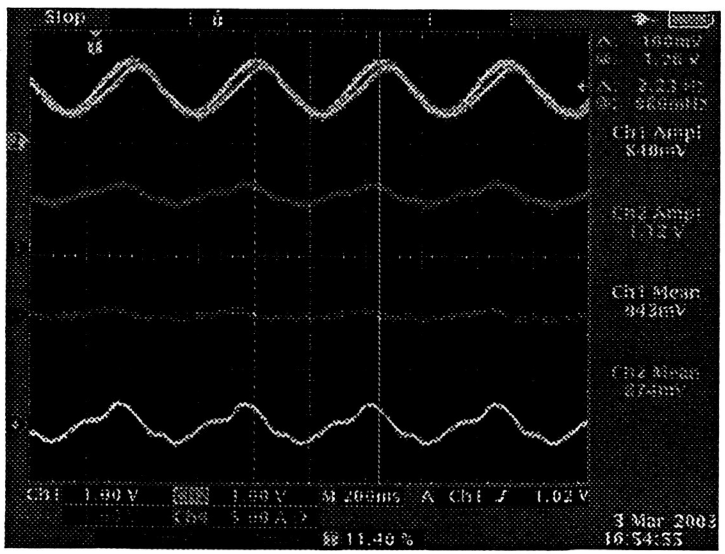
Evolution continued after fault using resulting population from the evolution of the original evolved controller design.

Marshall Space Flight Center
Jet Propulsion Laboratory

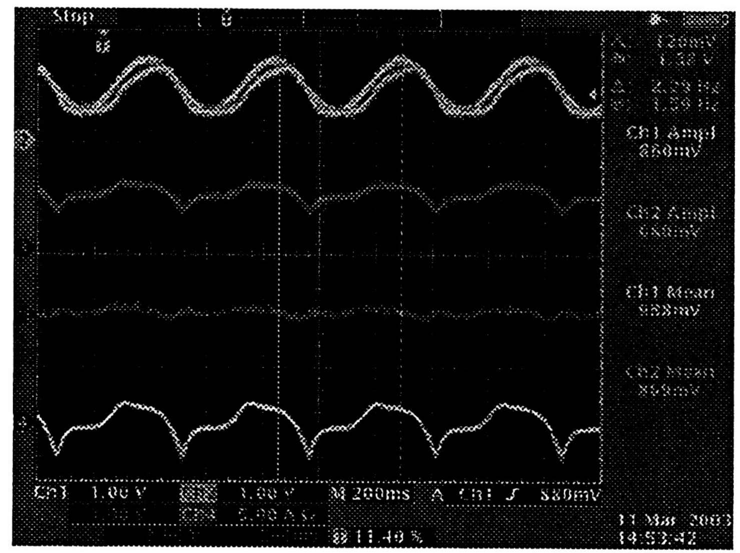
Reconfiguration for Fault Tolerance

Opening switch 19 or switch 71 causes output to flat line





Recovered response after 177 generations for switch 71

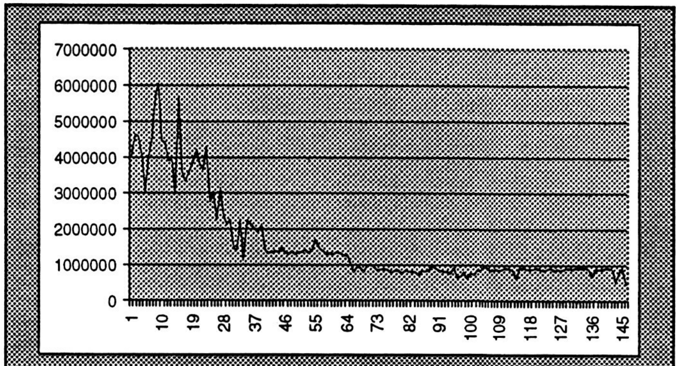


Recovered response after 146 generations for switch 19

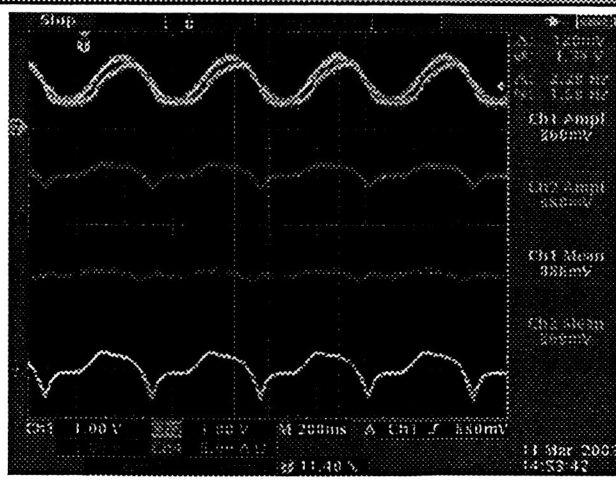
Reconfiguration for Fault Tolerance

Marshall Space Flight Center
Jet Propulsion Laboratory

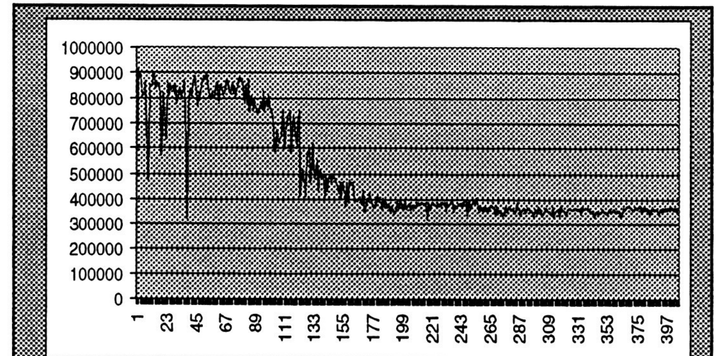
Extending evolution can provide improvements in recovered response



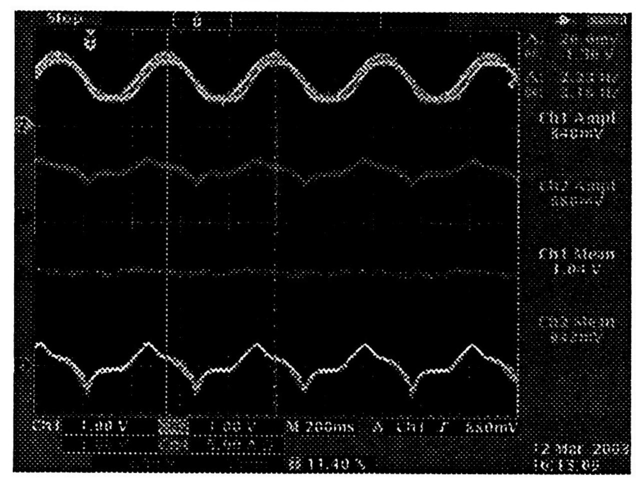
Fitness vs. Generation



Switch 19 recovered
Response before extension
(146 generations)



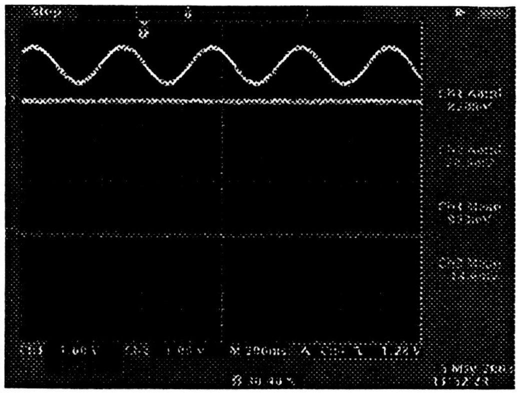
Fitness vs. Generation

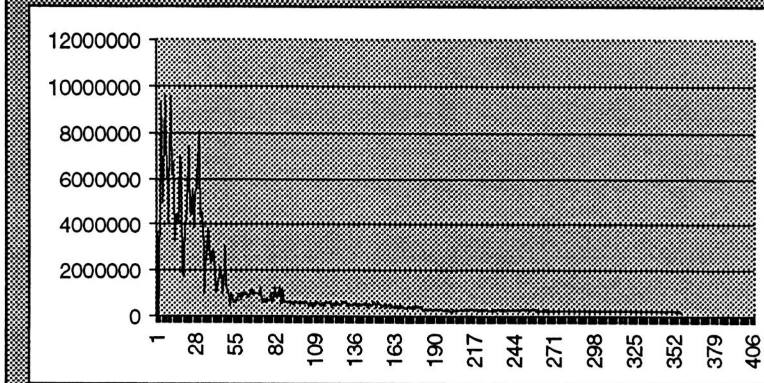


Recovered response after
406 further generations

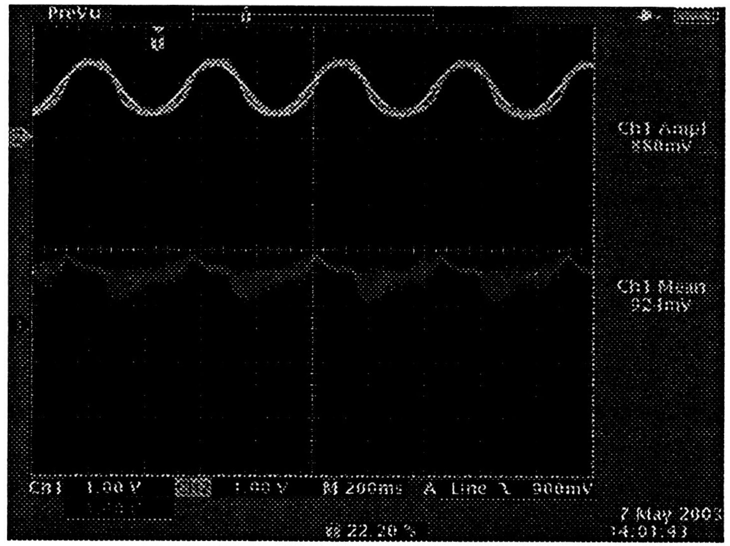
Reconfiguration for Fault Tolerance

Performance recovery with switches 71, 19 and 24 open





Fitness vs. Generation



Opening these three switches represents severe damage:

S71 & S19 cause output to flat line

S24 causes output switch between limits

Recovered response after 356 generations

Summary

- The results presented show the FPTA2 can be used to evolve simple analog closed-loop controllers that provide good response in comparison with a PI controller
- Evolution can be used to recover performance after faults
- Evolved design uses less transistors than the conventional design, and no external components.
  - Neglecting programmable overhead, considering functional components only
- Relatively slow response of many practical servomotor-driven systems is a time bottleneck that increases time to convergence
- Future work will include approaches for rapid and competent convergence to controller designs and more complex cascaded or multi-loop controllers
- Extension of fault tolerance experiments to external faults